Experiments on the Evolution of Digital to Analog Converters[1]

Ricardo Zebulum, Adrian Stoica, Didier Keymeulen
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
ricardo@brain.jpl.nasa.gov

Abstract – Evolvable Hardware (EHW) applications have, so far, encompassed the synthesis of standard analog and digital circuits' building blocks through Genetic Algorithms (GAs). Currently, the research effort in EHW is being driven towards twofold purposes: the synthesis of circuits of medium to high complexity; and the design of reconfigurable architectures that facilitate the system evolvability and on-chip implementation of the evolved circuits. This work addresses these issues by describing the evolution of Digital to Analog Converters (DACs). We investigate the efficiency of the evolutionary system when using different representations and when evolving current and voltage mode circuits. A new techinique based on hierarchical evolution is devised to enhance the evolutionary speed and the design scalability. New methods to increase the competitiveness of the evolved designs are also discussed.

TABLE OF CONTENTS

1. INTRODUCTION

This work describes a set of experiments involving the evolution of electronic circuits, particularly Digital to Analog converters. Several important issues for EHW are addressed by the experiments here presented, such as the evolution on reconfigurable architectures models, different modes of circuit operation, and the scalability of the results.

DACs have a widespread use in the area of digital audio, where the digital information stored in the CD is converted into music via high-precision converters. The synthesis of DACs includes several requirements, such as speed, linearity of the transfer curve, accuracy and resolution [1]. Most of the DACs presented in the literature are decoder based, in which a string of resistors is used to generate 2^N reference signals, where N is the number of digital inputs [2]. The digital input word is then used to select the appropriate reference signal. The evolutionary approach for circuit synthesis, on the other hand, usually explores alternate ways to synthesize circuits that are rather different from human designed ones.

The Field Programmable Transistor Array (FPTA) [3] is used as a reconfigurable hardware model for some of the evolutionary experiments. We contrast the results of evolution when using the FPTA model and another representation method previously presented in the literature [4].

This paper is divided into the following contents: section 2 provides a brief overview of Evolvable Hardware. Section 3 describes the Genetic Algorithm (GA) modeling for the DAC evolution. Section 4 depicts the experiments performed, where we compare the performance of the evolutionary system when using different circuit representations, and also for voltage and current mode circuits. Finally, section 5 concludes the work.

2. OVERVIEW OF EVOLVABLE HARDWARE

The conceptual birth of evolvable hardware was partially inspired by search/optimization/adaptation mechanisms and partially by the availability of reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) [5]. Circuits can be evolved via reconfiguring programmable devices (which is called *intrinsic* EHW) or via evolving software models – descriptions of the electronic HW (referred to as *extrinsic* EHW) [6].

[1] 0-7803-6599-2/01/$10.00 © 2001 IEEE

Figure 1 illustrates the main steps of evolutionary design for electronic circuits.

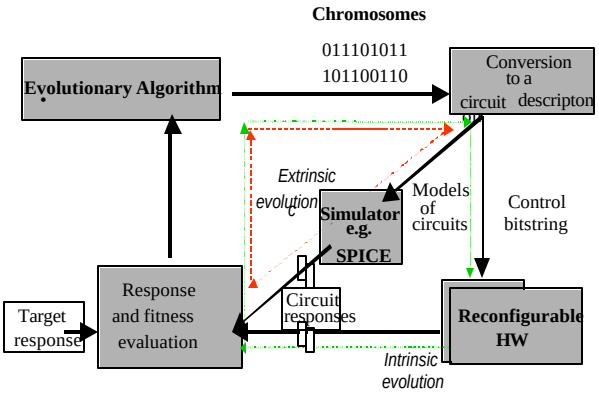


Figure 1- Evolutionary synthesis of electronic circuits

Each candidate circuit design is associated with a "genetic code" or chromosome. The simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encode a circuit. The first step of evolutionary synthesis is to generate a random population of chromosomes. In the case of extrinsic evolution, the chromosomes are then converted into a model that gets simulated (e.g. by a circuit simulator such as SPICE) and produces responses that are compared against specifications. In intrinsic evolution the chromosomes are converted into control bitstrings, which are downloaded to program the reconfigurable device. The configuration bitstring determines the functionality of the cells of the programmable device and the interconnection pattern between cells. Circuit responses are compared against specifications of a target response and individuals are ranked based on how close they come to satisfying it.

Many different trends can be currently identified in EHW research: evolution of analog circuits [7], exploration of device physics [8], fault tolerance experiments [9], evolution of digital circuits [10], industrial applications [11], and research on reconfigurable analog devices [12].

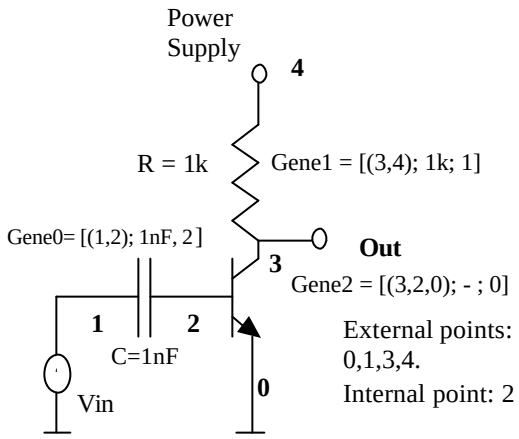
3. Problem Modeling

The evolution of DACs has been reported before in [13]. In this work, a 3 input bits DAC has been evolved using the Genetic Programming (GP) evolutionary technique [7]. GP allows a very flexible representation, in the sense that circuits of different sizes and shapes can be sampled by evolution. Particularly, Bennett et Al [13] used bipolar transistors, resistors and capacitors as the components to be manipulated by evolution. However, they needed to evaluate around 45,000,000 individuals (population of 330,000 individuals sampled along 139 generations) to achieve the solution. Another limitation of this experiment is the fact that the solution can not be implemented in hardware, because no integrated circuit model was employed to generate the software simulation models.

The problem modeling refers to the circuit *representation* and to the *fitness evaluation function*. This paper compares two different representations, the first one allowing an arbitrary pattern of interconnections among the components, which we call *gene based representation*, and the second one based on an existing reconfigurable chip.

The gene based representation establishes a straightforward mapping between the electronic circuit topology and the integer strings processed by the GA [4]. Each functional block of the string, also called gene, states the nature, value, and connecting points of a correspondent electronic component, which may include resistors, capacitors, bipolar transistors and MOS (Metal-Oxide-Semiconductor) transistors.

Figure 2 depicts an example of this kind of chromosome-circuit mapping for a common emitter amplifier. The chromosomes are made up of genes, each of which encodes a particular component. In the example of Figure 2, the chromosome will consist of three genes. The gene determines the nature, value and connecting points of the related component. The total number of connecting points is a parameter to be set in this representation. This parameter is critical to the efficiency of the representation: if too few connecting points are considered, the number of possible topologies sampled by the evolutionary algorithm will be limited; conversely, if too many connecting points are considered, a higher number of non-simulatable topologies (with floating components) will arise. Additionally, each connecting point may be classified as internal or external. While the former does not serve for any special purpose, the latter is connected to one of the following signals: power supply, ground, input signal or probed output (Figure 2).



Gene = [Connecting points, Component value,Component nature]. The Component nature is given by:

0 = transistor; 1 = resistor; 2 = capacitor

Figure 2– Analog Circuit Representation.

The advantage of this representation is its flexibility to map circuits with arbitrary types of interconnections. However, there is no integrated circuit model that support the hardware implementation of the evolved solutions.

The second representation utilizes a model for the simulation of a reconfigurable transistor array, the Field Programmable Transistor Array (FPTA). The FPTA is a concept design for hardware reconfigurable at transistor level. As both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA appears as a versatile platform for the synthesis of both analog and digital (and mixed-signal) circuits. The architecture is cellular, and has similarities with other cellular architectures as encountered in FPGAs (e.g. Xilinx X6200 family) or cellular neural networks. One key distinguishing characteristic relates to the definition of the elementary cell. The architecture is largely a "sea of transistors" with interconnections implemented by other transistors acting as signal passing devices (gray-level switches) and integrated on a chip using 0.5-micron CMOS technology. Figure 3 illustrates an FPTA cell consisting of 8 transistors and 24 programmable switches.

The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011…", where by convention one can assign 1 to a switch turned ON and 0 to a switch turned OFF. Programming the switches ON and OFF defines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation.

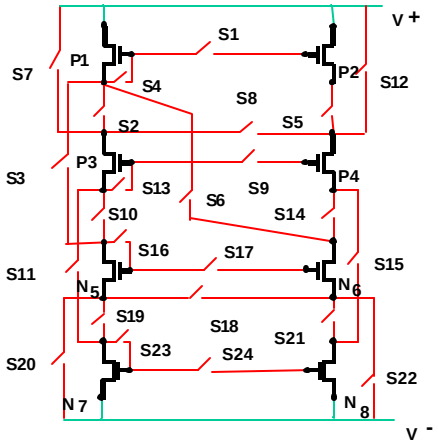


Figure 3 - Module of the Field Programmable Transistor Array.

The advantage of this second representation is the possibility of implementing in hardware the evolved solutions. However, the amount of different topologies potentially sampled by the GA is limited by the number of switches.

The *fitness evaluation function* determines how good each individual is considering a target to be achieved. In the case of DACs, each circuit is evaluated using a SPICE transient analysis. All the possible configurations of the input bits are given as fitness cases to the circuit, following an up-counter sequence. Figure 4 illustrates a typical response of a DAC circuit.

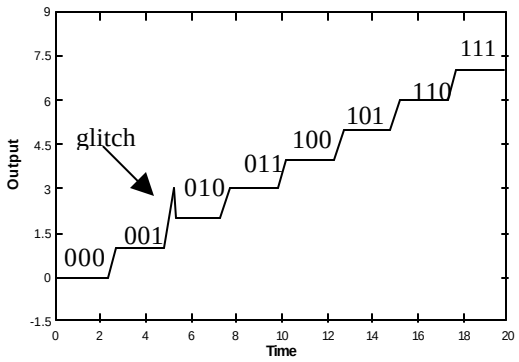


Figure 4 – Response of a Digital to Analog Converter.

The response in Figure 4 shows some of the non-idealities of DACs. The first one is the glitch problem, observed in the transition 001 to 010. It is a transient behavior in the output before it achieves the correct state. The second limitation is the *settling time*. It can be seen in the graph of Figure 4 that there is a time delay associated to the state transitions. The settling time must be kept in a value much lower than the signal frequency in order to be able to reconstruct the waveform [14].

Typically, the fitness evaluation function is measured by the weighted sum of the deviations between the desired and the actual responses:

$$Fitness = ? \; w_i \, |O_i - T_i| \qquad (1)$$

where O_i is the output sample at instant i, T_i is the associated target, and w_i is a weight. In this application, the weight was set to one if the absolute of the error was less than 0.5V, and 10 otherwise. This is the means whereby higher errors can be heavily penalized [13].

4. EXPERIMENTS

In this section three experiments are described. The first experiment employs the gene based representation for the evolution of a DAC operating in the voltage mode. The second experiment uses the FPTA model for the circuit representation, and operation in current mode. In the third experiment, the concept of hierarchical evolution is applied to the synthesis of a 4 bit DAC.

4.1 – First Experiment

In a first experiment, we have used the gene based representation to evolve a 3-bit DAC in voltage mode. Each chromosome was made up of ten genes, which could encode only for PMOS and NMOS transistors. In addition to determining the topological connections of the components

as illustrated in the previous sections, this representation also allowed another degree of flexibility, to determine the width (W) and length (L) of each transistor. Therefore, besides evolving the DAC topology, the GA performed a parametric optimization as well.

The parameters for the GA in the experiment were: 40 individuals, 100 generations, a crossover rate of 30% and mutation rate of 8%.

Two outputs of the circuit were considered for evaluation: the first one should provide the voltage variation resulting from the two least significant bits (LSB), and the second output must provide the voltage variation resulting from the most significant bit (MSB). These two outputs must be added to give the correct output for the 3-bit DAC. Figure 5 illustrates this concept. In this figure, *d0*, *d1* and *d2* represent the digital inputs, and *Out1* and *Out2* represent the circuit outputs. This is a divide-and-conquer approach devised to improve the performance of the evolutionary system.

A critical problem in the performance evaluation refers to the timescale in which the input bits change. If the timescale is small (milliseconds or less), than we may evolve a transient circuit, in which the transient response is correct, but that achieves an erroneous permanent state. This problem is depicted in Figure 6, which shows the results of a DAC circuit evolved in a preliminary experiment. This particular circuit was evaluated using a timescale of milliseconds, but we can observe the change in behavior when we increase the timescale: the final state reached for the inputs 100 and 101, which appeared to be right in the first simulation (graph A, spanning 20ms), is wrong in reality (graph B, spanning 20 seconds). In order to overcome this problem, in this new evolutionary experiment each circuit was evaluated twice, using timescales of milliseconds and seconds.

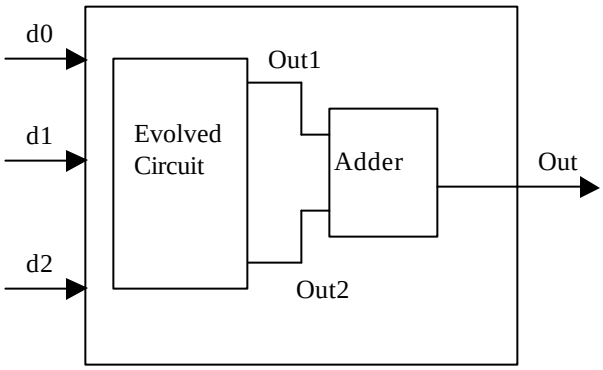


Figure 5 – Divide and Conquer approach for the evolution of the Digital to Analog Converter.

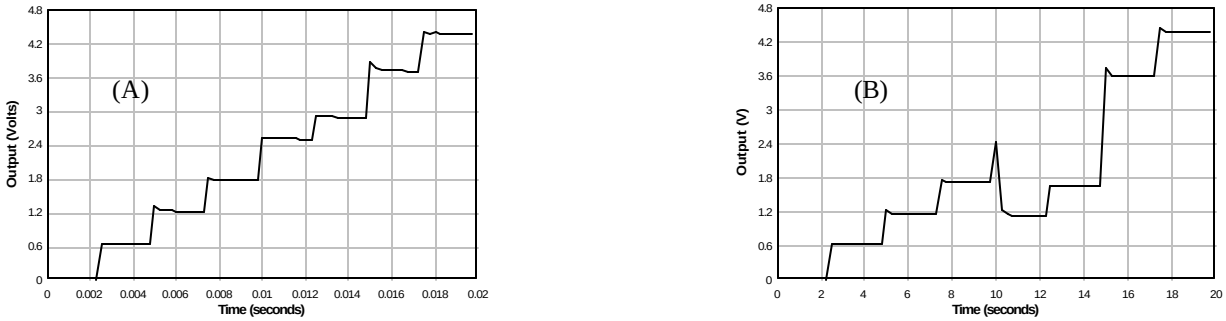


Figure 6 – 3-bit DAC evolution using a timescale of milliseconds (A) and the erroneous behavior of the circuit when tested in a timescale of seconds (B).

Figure 7 depicts the schematic of the evolved circuit using the new evaluation approach, and Figure 8 shows the responses: the curves of the two outputs are shown in the first graph, and the combined output is shown in the second graph. In the circuit of Figure 7, the substrate of NMOS transistors are connected to ground, and substrate of PMOS transistors are connected to 5 Volts. The evolved length and width of the transistors are given as L/W in um.

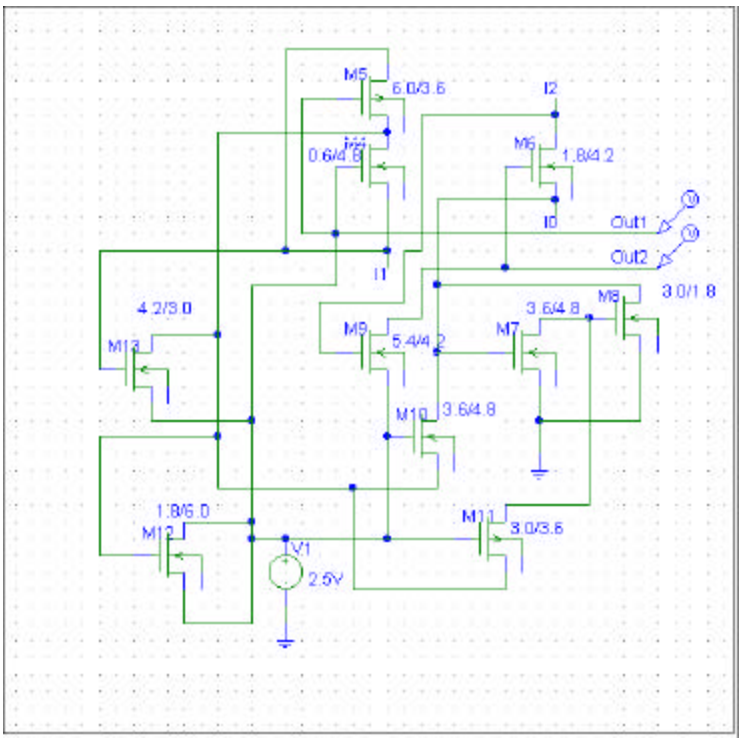


Figure 7 – Evolved Digital to Analog Converter. Inputs are labeled as I0, I1 and I2. Outputs are labeled as Out1 and Out2.

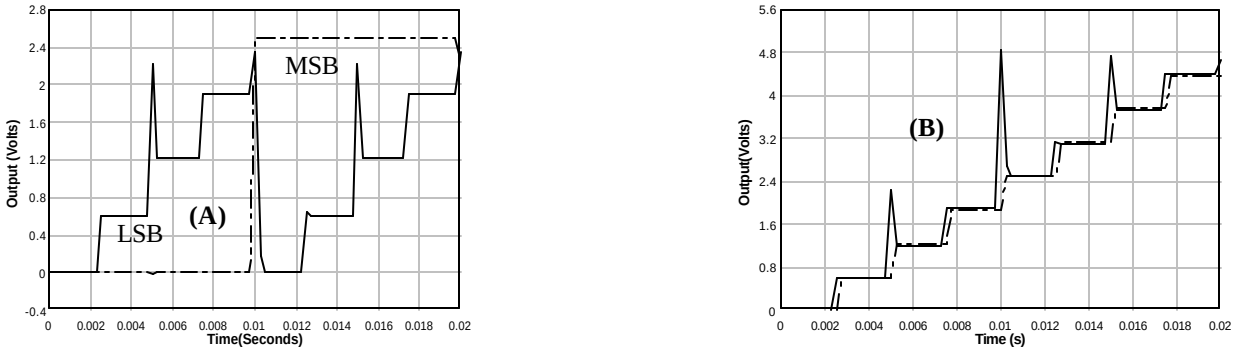


Figure 8 – Two outputs, LSB and MSB, of the evolved circuit in full lines and in traces respectively (A) and combined output in full lines against target in traces (B).

It can be seen from Figure 8.B that the evolved circuit achieves a final response practically equal to the target for each of the eight states. Nevertheless, the circuit suffers fromthe glitch problem previously discussed.

4.2 – Second Experiment

The second experiment aimed to improve the evolvability of the circuit.

A binary string was used to represent each individual, mapping two cascaded FPTA cells (Figure 3). Two binary inputs are applied to the circuit and a current is measured at the output.

In order to facilitate the evolvability, we relaxed the requirement of having a specific DC operating point for each state of the converter. Instead, the fitness evaluation function rewarded circuits presenting uniform stepwise output. The following measure has been employed:

$$\text{Fitness} = \langle ?I_i \rangle - K \cdot \text{MSE}(?I_i, \langle ?I_i \rangle) \quad (2)$$

For each state transition *i*, we measure the gradient in the current output $?I_i$. The average $<?I_i>$ and Mean Squared Error to the average value, $MSE(?I_i, <?I_i>)$, are computed over the three state transitions. The constant *K* is a weighting factor determined by the user. This fitness measure seeks to maximize the current gradient between states, also keeping them uniform for all the transitions. The fact that the circuit is operating in current mode allows this degree of flexibility to be incorporated into the fitness function. Instead of specifying the particular values of the current at each state, only the shape of the output is evolved.

The GA sampled only 30 individuals along 30 generations to find a circuit, since this is a very simple task for evolution. The evolved circuit is shown in Figure 9 and its response is depicted in Figure 10.

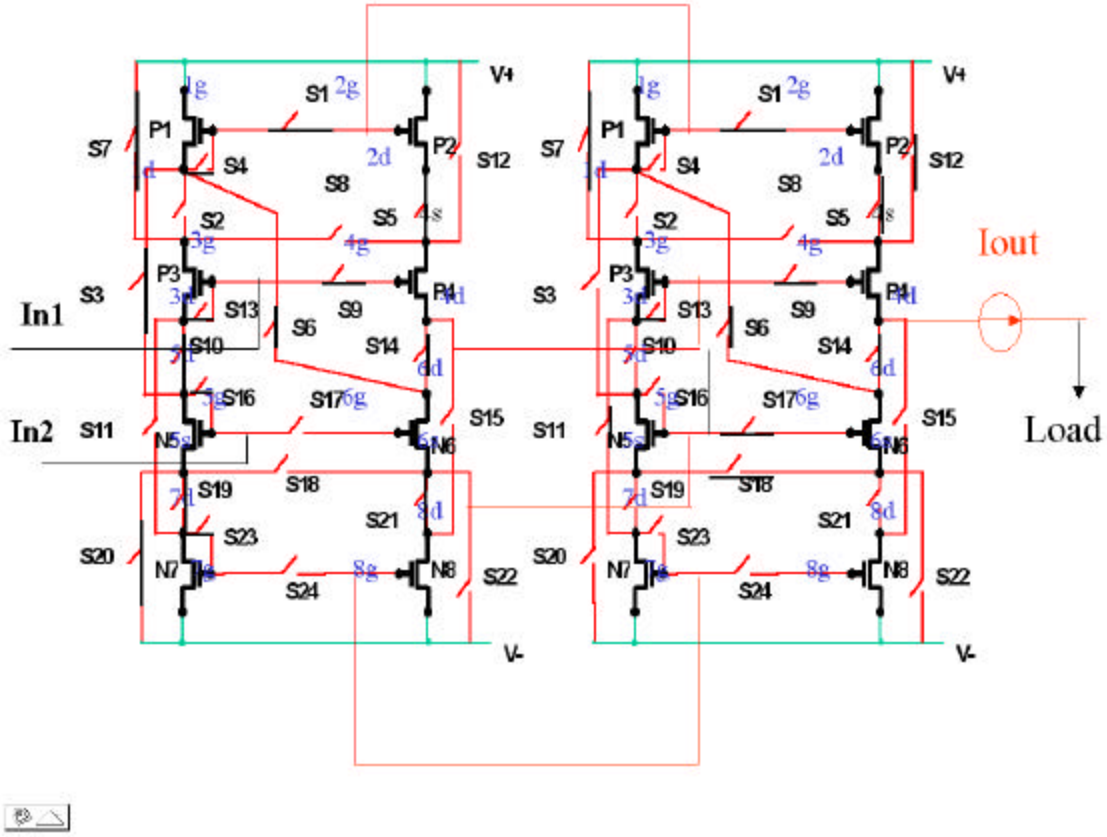


Figure 9 – Two bit Digital to Analog Converter evolved in the second set of experiments.

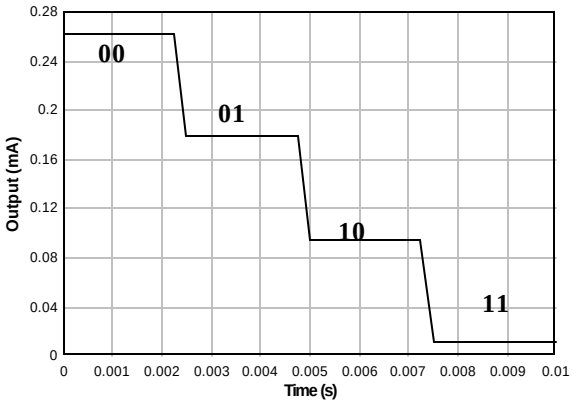


Figure 10 – Response of the circuit shown in Figure 9. (Current in mA x Time in seconds).

Besides being a quick and easy task for the GA, the evolution of current mode circuits facilitates the design scalability. For instance, we can easily construct a 4-bit DAC using the circuit of Figure 9 as a building block. As shown in Figure 11, all that is needed is a pair of current mirrors [1] with ratios of 1:1 and 1:4 respectively. Figure 12 shows the response of the 4-bit DAC.

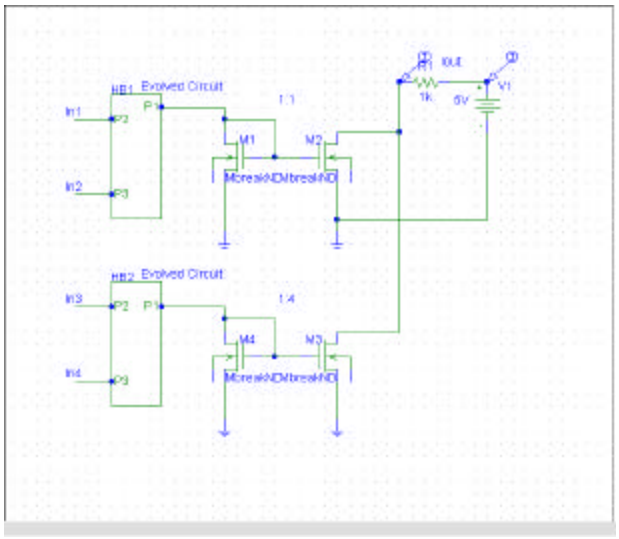


Figure 11 – 4 bit DAC based on evolved building block of Figure 9.

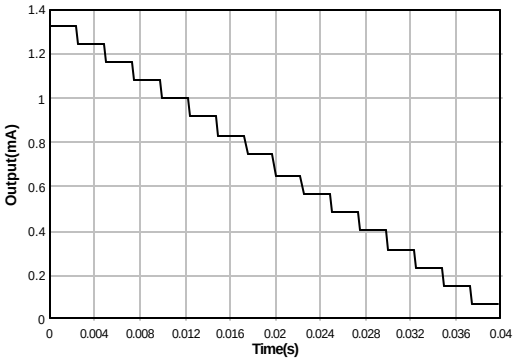


Figure 12 – Output of the circuit shown in Figure 11.

4.3 – Third Experiment – Hierarchical Evolution

The third experiment also targeted the synthesis of a 4-bit DAC. However, contrasting to the experiment shown above, evolution will also assemble the building blocks. In the circuit of Figure 11, human knowledge has been used to achieve a 4 bit DAC using current mirrors and the 2-bit evolved DAC. This experiment allows evolution to manipulate the building blocks, without human intervention. The following building blocks are used:

?? Current mirrors of different gains (1/8, 1/4, 1/2, 1, 2, 4, 8);
?? The evolved 2-bit DAC of Figure 9.

This concept was first applied to evolve a 3-bit DAC. Figure 13 depicts the evolved circuit and Figure 14 its response.

We proceeded one step further to achieve the 4-bit DAC. Evolution was now allowed to use the 3-bit DAC as a building block, together with the current mirrors. Figure 15 depicts the evolved circuit.

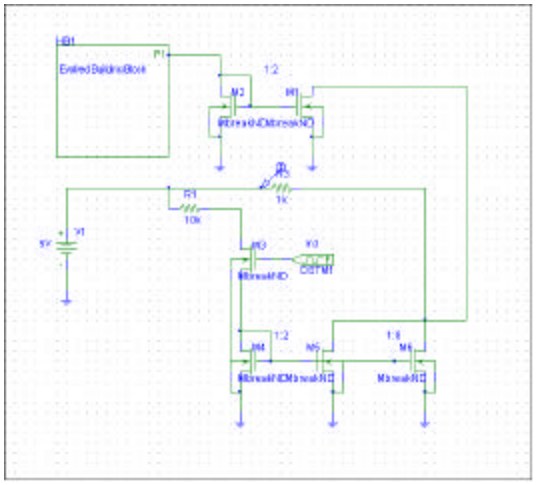


Figure 13 – Evolved 3-bit DAC without human intervention. *In3* is the MSB.

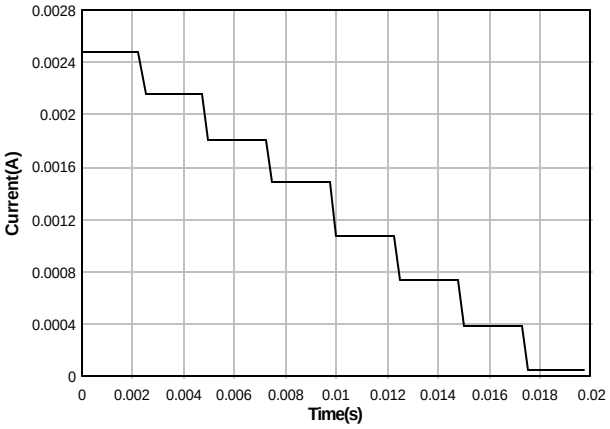


Figure 14 – Output of the circuit shown in Figure 13.

The fitness described by equation (2) was also used in this experiment. The graph of Figure 16 displays the fitness along the 800 generations comprised by this experiment.

We remark that, since evolution now manipulates high level building blocks, it is not necessary to use SPICE to simulate the circuits generated by the evolutionary process. The DC operating point of each building block has already been defined, and their behavior can be simulated using a high level description language, such as C. The main advantage of this procedure is the dramatic speed up in evolution time, taking less than one minute in a SPARC Ultra 2 Sun workstation.

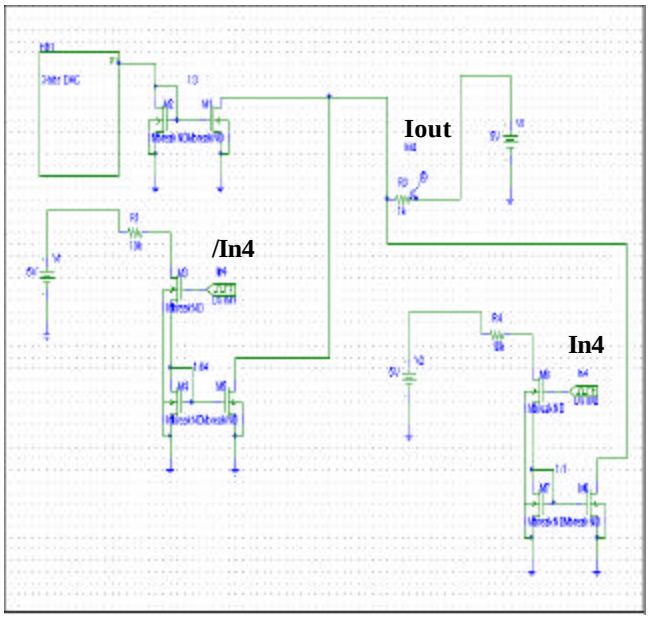


Figure 15 – Evolved 4-bit DAC. *In4* is the MSB, */In4* is the complemented version of the MSB and the output is I_{out}.

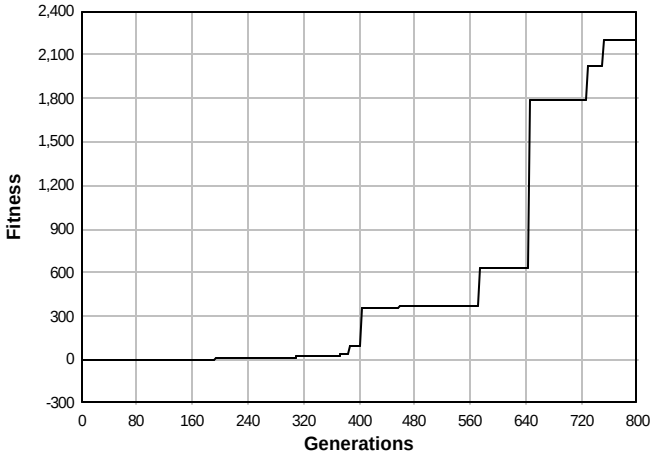


Figure 16 – Fitness behavior in the 4-bit DAC evolutionary experiment.

Figure 17 shows the response of this circuit. One performance criteria for DACs is the Differential Nonlinearity (DNL)[1]. This statistics can be defined as:

DNL_i = Incremental height of transition i – Ideal Increment Height (3)

In this particular case, there are 15 transitions, hence 15 different DNL values. The ideal incremental height of each transition is the LSB (Least Significant Bit) of the data converter, which is around 1 mA in this case. The graph of Figure 18 plots the DNL as a percentage of the LSB for all the transitions. The worst case occurs at the 8th transition, in which the DNL is around 0.5 (50% higher than the LSB). This transition is marked in the graph of Figure 17.

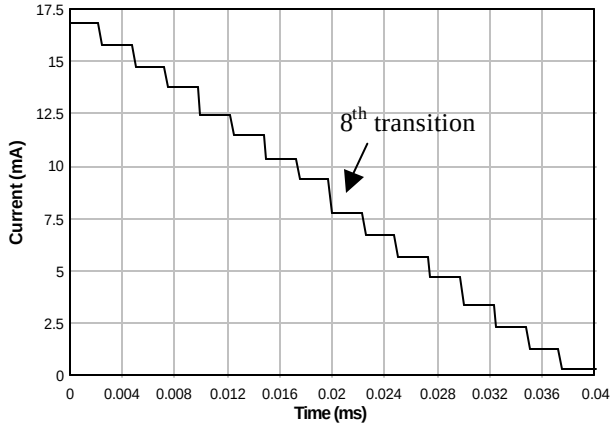


Figure 17 – Response of the circuit shown in Figure 15.

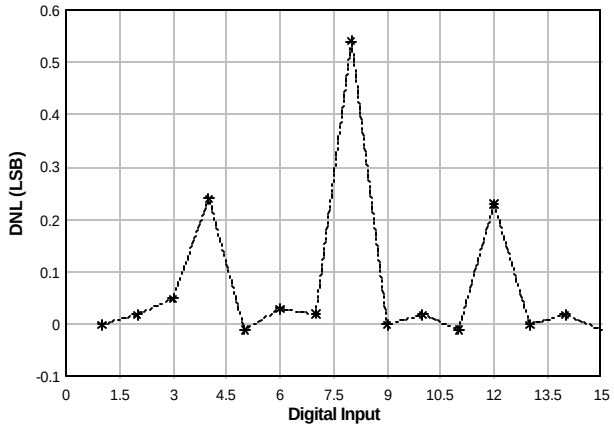


Figure 18 – DNL for the circuit shown in Figure 15.

In order to reduce the DNL, a parametric optimization has been performed, by making slight changes in the transistors sizes of the current mirrors. The response of the new circuit is shown in Figure 19, and the DNL statistics is shown in Figure 20. It can be observed that the highest DNL is now only 0.11, compared to 0.5 in the previous version of the circuit.

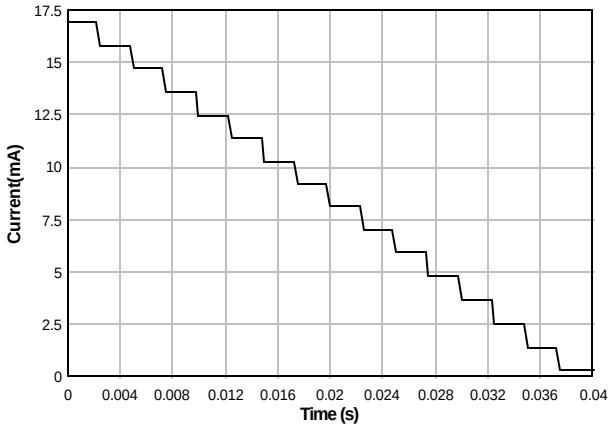


Figure 19 – Response of the optimized version of the circuit shown in Figure 15.

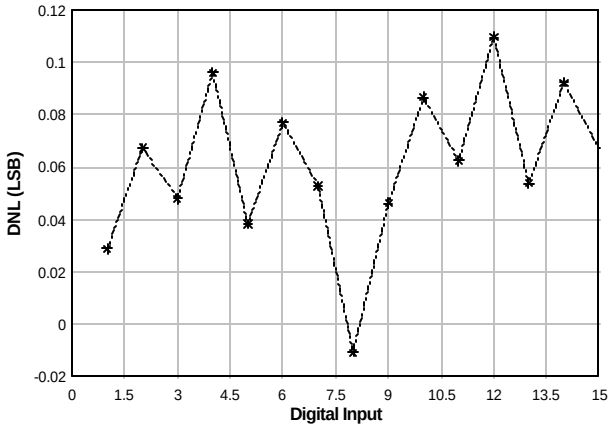


Figure 20 – DNL for the optimized version of the circuit shown in Figure 15.

Another performance statistics for DACs is the *settling time* [14], which is the time between the switching of the digital inputs of the converter and the time when the output reaches its final value and remains within a specified error band. The settling time must be faster than the signal frequency in order to be able to reconstruct the waveform. In this evolved circuit, the worst settling time (0111? 1000 and 1111? 0000 transitions) was of 20ns; hence this DAC can be used in applications involving frequencies of up to 50 MHz.

The main advantages of this approach are the scalability and the speed of evolution. Although we stopped at the 4-bit DAC, we can scale up the process to include more bits into the circuit input.

Up to some extent, this circuit still suffers from the glitch problem observed in the first experiment. There is a glitch in the transition 0111? 1000, too narrow that it does not appear in the graphs. However, this glitch is much smaller than the ones observed in Figure 8.

5. LESSONS LEARNED

The automatic synthesis through evolutionary means of Digital to Analog converters has been demonstrated in this paper. Two different representations were employed for circuit evolution, the gene based and the FPTA representation. While the former allows a more flexible pattern of interconnections, circuits evolved using the FPTA model can be implemented in hardware using an existing reconfigurable chip. It has been observed that the FPTA model provided better results for current mode circuits, while the gene based representation produced better results for voltage mode DACs.

A divide-and-conquer approach has been used to evolve a voltage mode DAC. Two circuit outputs were sampled, and later added in a post-processing stage. The steady state response of the circuit is very accurate comparing to the target, however the circuit suffers from glitch problem.

It has also been verified that caution should be taken referring to the timescale of the inputs. The circuits should be tested in different timescales, to consider both transient effects and the steady state response.

The last experiment consisted of the hierarchical evolution of a 4-bit DAC. In this particular case, the evolutionary system manipulated higher level building blocks, such as the evolved 2-bit DAC and current mirrors with different gains. The main advantages of hierarchical evolution are the speed and scalability. Since the building blocks can be described in a higher level description language, SPICE simulations are not necessary, and the evolution time drops to less than one minute. The design can be scaled up if we use circuits evolved in previous steps as building blocks for the next evolutionary step, which encompasses the evolution of larger circuits. The final 4-bit DAC obtained through this approach showed good statistics in terms of differential non-linearity and settling time.

The main problem of the evolved circuits is the glitches observed in the state transitions. This same problem has also been reported in [13], where GP is applied to evolve a 3-bit DAC. In order to solve this problem, the authors of the referred paper tuned the fitness function to sample the so-called tum-defining points internally generated by SPICE. Those are the regions where the glitches may occur.

Our approach to overcome this drawback is to use the GA to further optimize the transistors width and length of the evolved topologies. This approach has been successfully applied in this paper to improve the differential non-linearity of the DAC circuit. In this new GA run, the topology is kept fixed, enabling the GA to focus solely on parametric optimization. Moreover, the fitness evaluation

function will be designed in such a way to explicit penalize the glitches, by sampling the data points in the state transitions.

Future experiments will include multi-objective optimization, in order to provide circuits with superior performance compared to the human designed ones. The idea here is to decompose the fitness evaluation to consider the following factors:

- ?? Transient response with a small time step, minimizing glitches and settling time;
- ?? Steady state response;
- ?? Power dissipation;
- ?? and intrinsic noise.

There are many techniques proposed in the literature to carry out multi-objective optimization. The Energy Minimization technique [4] will be employed in these new experiments. The main goal is to achieve low-power and low-noise DACs of interest for aerospace systems.

6 - ACKNOWLEDGEMENT

The work described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the Defense Advanced Research Projects Agency and by the National Aeronautics and Space Administration.

REFERENCES

[1] – R. Baker, H. Li, D. Boyce, "CMOS Circuit Design, Layout, and Simulation", IEEE Press, 1998.

[2] – D. Johns, K. Martin, "Analog Circuit Integrated Design", John Wiley & Sons, 1997.

[3] - R. Zebulum, A. Stoica and D. Keymeulen, "A flexible model of a CMOS field programmable transistor array targeted for hardware evolution", Third Int. Conference on Evolvable Systems: From Biology to Hardware (ICES2000), Edinburgh, April 17-19, 2000, 274-283.

[4] - R. Zebulum, M.A. Pacheco, M. Vellasco, and H. T. Sinohara, "Evolvable Hardware: Automatic Synthesis of Analog Control Systems". In *IEEE Aerospace Conference*, Big Sky, Montana, March 14-25, 2000. IEEE Press.

[5] – E. Sanchez, "Filed Programmable Gate Array (FPGA) Circuits", in *Towards Evolvable Hardware: The evolutionary engineering approach*, pp. 1-18, E. Sanchez and M. Tomassini (editors), Springer-Verlag LNCS 1062, 1996.

[6] – A. Stoica, R. Zebulum, D. Keymeulen, Y. Jin, T. Daud, "Evolutionary Design of Smart Analog Circuits" to appear in the Proceedings of ANNIE'2000 (Smart Enginering System Design), St. Louis, MO, Novenber 5-8, 2000.

[7] - J. R. Koza, F. H. Bennett III, D. Andre and M. A. Keane, "Genetic Programming III – Darwinian Invention and Problem Solving", Morgan Kaufman, San Francisco, 1999.

[8] – A. Thompson., P. Layzell., R. Zebulum, "Explorations in Design space: Unconventional Electronics Design Through Artificial Evolution", IEEE Trans. on Evolutionary computation, vol. 3, n. 3, pp. 167-196, 1999.

[9] - D. Keymeulen, A. Stoica, R. Zebulum, "Fault-Tolerant Evolvable Hardware using Field Programmable Transistor Arrays". In *IEEE Transactions on Reliability, Special Section on Fault-Tolerant VLSI Systems*, vol. 49, No. 3, 2000 Sept. IEEE Press.

[10] – J. Miller, P. Thomson, "Aspects of Digital Evolution: Geometry and Learning", in Proceedings of the Sec. Int. Conf. on Evolvable Systems. M.Sipper, D.Mange and A. Pérez-Uribe (editors), vol. 1478, pp. 25-35, LNCS, Springer-Verlag, 1998.

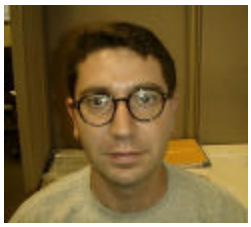
[11] – T. Higuchi et al, "Real-World Applications of Analog and Digital Evolvable Hardware". In IEEE Transactions on Evolutionary Computation, Vol.3, No. 3, September 1999. IEEE Press.pp.220-235.

[12] - A. Stoica, "Toward evolvable hardware chips: experiments with a programmable transistor array". Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, April 7-9, IEEE Comp Sci. Press, 1999. pp 156-162.

[13] – F. H. Bennett III, J. R. Koza,, M. A. Keane, J. Yu, W. Mydlowec, and O. Stiffelman, "Evolution by means of genetic programming of analog circuits that perform digital functions". In Banzhaf, Wolfgang, Daida, Jason, Eiben, A. E., Garzon, Max H., Honavar, Vasant, Jakiela, Mark, and Smith, Robert E. (editors). 1999. GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, 1999, Orlando, Florida USA. San Francisco, CA: Morgan Kaufmann. Pages 1477 - 1483.

[14] – R. Mancini, C. Wray, " Analog Electronics in a Day", WESCON'2000, Tutorial, September 19-21, Anaheim, CA.

Dr. Ricardo Zebulum
Jet Propulsion Laboratory, M/S 303-300, California Institute of Technology, Pasadena, California, 91109 USA.
Email: ricardo.zebulum@jpl.nasa.gov



Ricardo S. Zebulum is a post-doc scholar at the Jet Propulsion Laboratory, California Institute of Technology. He received his Bachelor degree in Electronic Engineering in 1992, his Msc. in Electrical Engineering in 1995, and his PhD in Electrical Engineering in 1999, all of them at the Catholic University of Rio, Brazil. He stayed two years at Sussex University, from 1997 to 1999, as a visiting PhD student. He has been involved with research in Evolvable Hardware since 1996. His research interests also include artificial vision systems, fault-tolerant electronics, low power electronics, MEMs, electronics for extreme environments and analog VLSI design.

Dr. Adrian Stoica
Jet Propulsion Laboratory, M/S 303-300, California Institute of Technology, Pasadena, California, 91109 USA.
Email: adrian.stoica@jpl.nasa.gov

Adrian Stoica is a Senior Researcher in the Advanced Computing Technologies Group at NASA's Jet Propulsion Laboratory (JPL), Pasadena, CA. He is leading the JPL research in Evolvable Hardware. His current NASA/JPL projects are in the areas of Evolvable Hardware, Sensor Fusion Hardware, Rover Intelligence, and Robot Fostering. He is also the Principal Investigator for the Evolvable Hardware for Adaptive Computing project funded by DARPA. Dr. Stoica's research is directed along two themes: adaptive hardware for autonomous space systems, and next-generation robots. Dr. Stoica received a M.S. degree in Electrical Engineering from the Technical University of Iasi, Romania, in 1986, and a Ph.D. in Electrical Engineering and Computer Science from Victoria University of Technology in Melbourne, Australia, in 1996. In 1999 he organized the First NASA/DOD Workshop on Evolvable Hardware, initiating a series of U.S. meetings dedicated to this field.

Dr. Didier Keymeulen;
Jet Propulsion Laboratory, M/S 303-300, California Institute of Technology, Pasadena, California, 91109 USA.
Email: didier.keymeulen@jpl.nasa.gov

Didier Keymeulen received his PhD in Electrical Engineering and Computer Science from the Free University of Brussels. Before joining JPL in 1998, he worked at the National Electrotechnical Laboratory in Japan on the applications of evolvable hardware for robotics. At JPL, he is responsible for the applications of evolvable hardware to fault-tolerant electronics and adaptive sensor technology. His interests include adaptive hardware for embedded systems.